IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

EXHIBIT A

Excerpts from

Held, Gilbert, Voice and Data Internetworking

For Serial No.: 09/519,605
Applicant(s): SUN, Peter

Voice and Data Internetworking

Gilbert Held

McGraw-Hill
New York • San Francisco • Washington, D.C. • Auckland • Bogotá
Caracas • Lisbon • London • Madrid • Mexico City • Milan
Montreal • New Delhi • San Juan • Singapore
Sydney • Tokyo • Toronto

Library of Congress Cataloging-in-Publication Data

Held, Gilbert.
Voice and data internetworking with IP and frame relay / Gilbert Held.
p. cm. — (McGraw-Hill series on computer communications)
Includes index.
ISBN 0-07-028135-1
1. TCP/IP (Computer network protocol) 2. Digital telephone systems. 3. Computer networks. 4. Data transmission systems.
I. Title. II. Series.
TK5105.585.H45 1998
621.385—dc21 98-6456
CIP

McGraw-Hill
A Division of The McGraw-Hill Companies

1 2 3 4 5 6 7 8 9 0 AGM/AGM 9 0 4 3 2 1 0 9 8

ISBN 0-07-212216-1

The sponsoring editor for this book was Steven Elliot, and the production supervisor was Clare Stanley. It was set in Vendome by North Market Street Graphics.

Printed and bound by Quebecor/Martinsburg

turn our attention to the format of their headers to include their port number field, which is used in conjunction with IP address fields by routers and firewalls as a mechanism to filter packets.

The TCP Header

At the transport layer, TCP accepts application data in chunks of up to 64 Kbytes in length. Those chunks are fragmented into a series of smaller pieces that are transmitted as separate IP datagrams, typically 512 or 1024 bytes in length. Since IP provides no mechanism that guarantees datagrams will be correctly received as to both content and sequence, it is up to the TCP header to provide the mechanism for reliable and orderly delivery of data. To do so, the TCP header includes a field that is used for the sequencing of datagrams and a checksum field for reliability. Because traffic from different applications, such as FTP and HTTP, can flow from or to a common host, a mechanism is required to differentiate the type of data carried by each datagram. This data differentiation is accomplished by the use of a destination port field containing a numeric that identifies the process or application in the datagram. In actuality, the TCP header plus data is referred to as a *segment*, so the port number identifies the type of data in the segment, and the IP header is added to the TCP header to form the datagram that will contain the source and destination IP address. Now that we have a general appreciation for the TCP header and its relationship to the application process and IP header, let's turn our attention to the fields in the TCP header whose structure is illustrated in Figure 2-21.

Source and Destination Port Fields

The source and destination port fields are each 16 bits in length. Each field identifies a user process or application, with the first 1024 out of 65,536 available port numbers standardized with respect to the type of

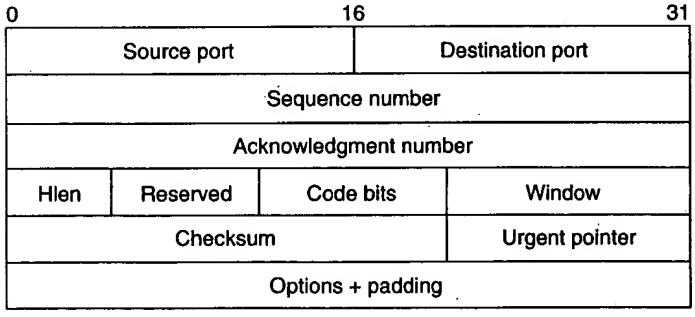

Figure 2-21
The TCP header.

traffic transported via the use of a specific numeric value. The source port field is optional and, when not used, is set to a value of 0. The term *well-known port*, which is commonly used to denote an application layer protocol or process, actually refers to a port address at or below 1023. Both TCP and UDP headers contain fields for identifying source and destination ports. For example, Telnet, which is transported by TCP, uses the well-known port number 23, while SNMP, which is transported by UDP, uses the well-known port number 161.

Sequence and Acknowledgment Number Fields

The sequence number field is 32 bits in length and provides the mechanism for ensuring the sequentiality of the data stream. The acknowledgment number field, which is also 32 bits in length, is used to verify the receipt of data.

Hlen Field

The Hlen field is 4 bits in length. This field contains a value that indicates where the TCP header ends and the data field starts. This field is required because the inclusion of options can result in a variable-length header.

Code Bits Field

The code bits field is also referred to as a *flags field*, as it contains 6 bits, each of which is used as a flag to indicate whether a function is enabled or disabled. Two bit positions indicate whether or not the acknowledgment and urgent pointer fields are significant. The purpose of the urgent bit or flag is to recognize an urgent or a priority activity, such as when a user presses the CTRL-BREAK key combination. Then the application will set the Urgent flag, which results in TCP immediately transmitting everything it has for the connection. The setting of the urgent bit or flag also indicates that the urgent pointer field is in use. Here, the urgent pointer field indicates the offset in bytes from the current sequence number where the urgent data is located. Other bits or flags include a PSH (push) bit, which requests the receiver to immediately deliver data to the application and forgo any buffering, an RST (reset) bit to reset a connection, a SYN (synchronization) bit used to establish connections, and a FIN (finish) bit, which signifies the sender has no more data and the connection should be released.

Window Field

The window field is 2 octets in length. This field is used to indicate the maximum number of blocks of data the receiving device can accept. A

large value can significantly improve TCP performance, as it permits the originator to transmit a number of blocks without having to wait for an acknowledgment and permits the receiver to acknowledge the receipt of multiple blocks with one acknowledgment. Although each field in the TCP header is important, the goal of this chapter is to provide an understanding of the operation of voice over IP and the configuration of equipment required to support it, so we will not probe deeper into the TCP header. Instead, we will examine the UDP header and conclude the chapter by discussing the Real-Time Transport Protocol, the Resource ReSerVation Protocol (RSVP), and the H.323 standard.

The UDP Header

Through the use of UDP, an application can transport data in the form of IP datagrams without having to first establish a connection to the destination. This also means that when transmission occurs via UDP, there is no need to release a connection, which simplifies the communications process. This in turn results in a header that is greatly simplified and much smaller than TCP's header.

Figure 2-22 illustrates the composition of the UDP header, which consists of 16 bytes followed by actual user data. Similarly to TCP, an IP header will prefix the UDP header. The resulting message, consisting of the IP header, the UDP header, and user data, is referred to as a UDP *datagram*.

Source and Destination Port Fields

The source and destination port fields are each 2 octets in length and function in a similar manner to their counterparts in the TCP header. That is, the source port field is optional and filled with 0s when not in use, while the destination port contains a numeric that identifies the application or process. Since UDP is commonly used by several Internet telephony products, you must determine the port a specific product uses. Then you will probably have to reprogram your organization's router access list and modify the configuration of your organization's firewalls to enable UDP datagrams using ports previously blocked to transport Internet telephony data onto your private network via the Internet.

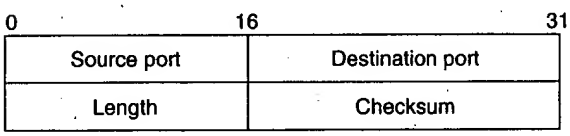

Figure 2-22
The UDP header.